Analysis on Romance Movies with the IMDb dataset

PIC 16 Final Project

Presented by Sihan Min,
Jinghui Song,
Junhong Shen

OUR PROCESS

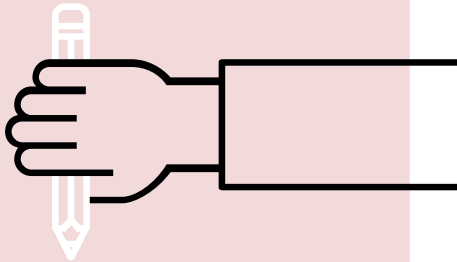
I:
Trend Analysis
Years
&
Popularity

II:
Visualization
Romance Movies
&
Production
Countries

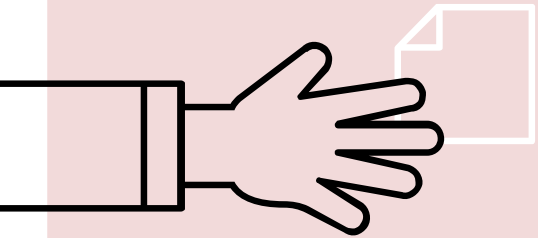
III:
Classification
Romance Movies
&
Movie Quality

Idea

- Use a **subset** of data -> obtain specific conclusions
- Explore the **internal pattern** of romance movie production
- **Generalize** to more romance movies



I. Popularity Over Time

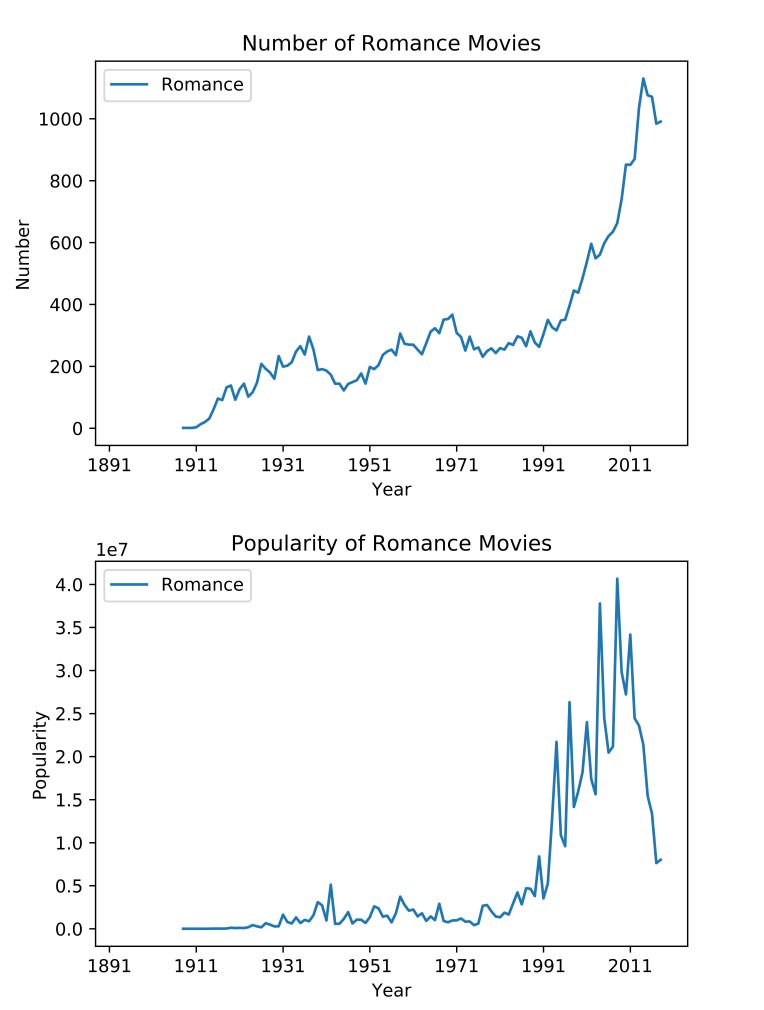


Basic Data Cleaning

- **Original Data**: (downloaded at https://datasets.imdbws.com)
  - title.basics.tsv.gz (tconst, titleType, startYear, genres)
  - title.ratings.tsv.gz (tconst, averageRating, numVotes)
  - title.akas.tsv.gz
- Use **pandas**.DataFrame.apply and **filter functions** to find all the movies before 2019
- **Merge** with ratings, and **compute popularity** (averageRating * numVotes) for each movie
- Basics: 5687200 x 9 Ratings: 914862 x 3
- Movies: 450481 x 4 (tconst, startYear, genres, Popularity)

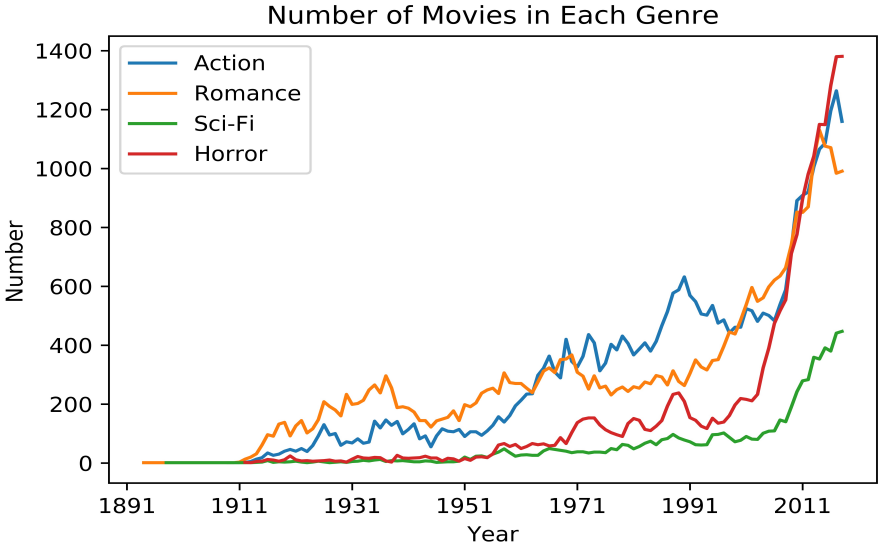
Data Aggregation

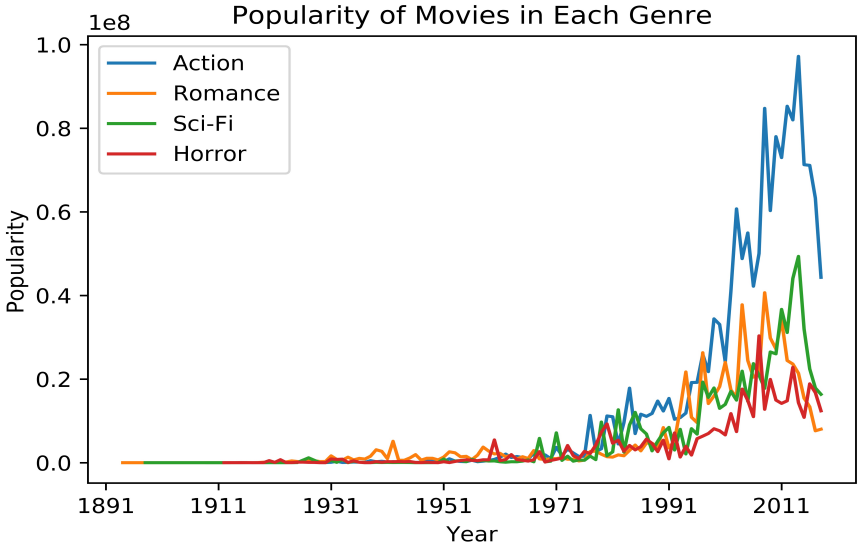
- Goal: IMDb **user preference** on movie genres over time
- Use count() and sum() in **pandas.DataFrame.groupby** on **'startYear'**
- Use df.apply and filter functions to find specific genre
- **Two Measurements:** total number and popularity



Genre Comparison

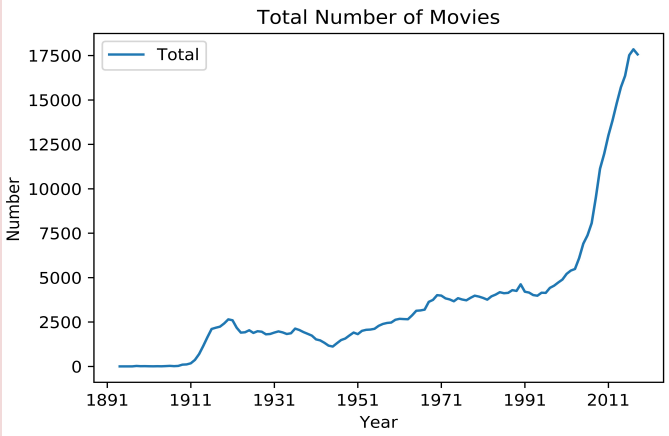
- Four genres: Romance, Action, Science Fiction, Horror

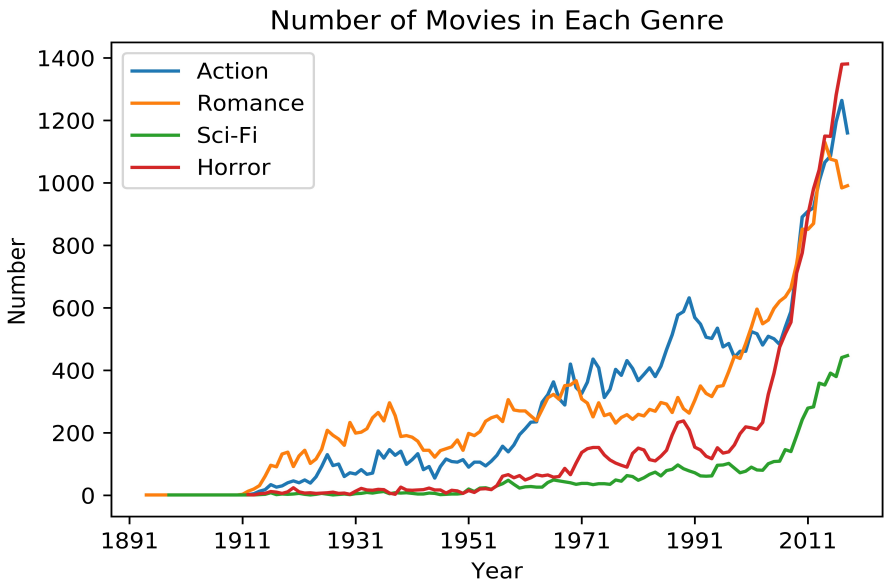


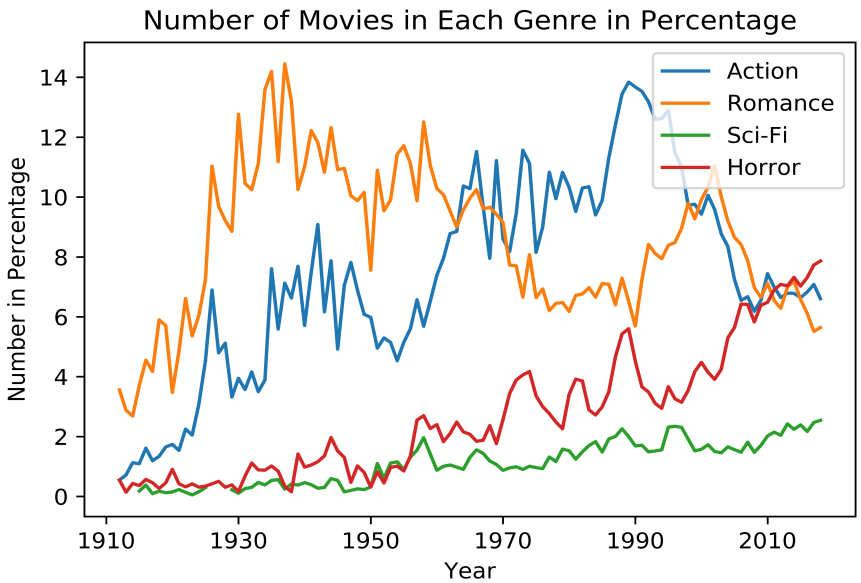


Regularization

- **Problem**: The movie industry is growing itself
- **Solution**: Take percentage on movie numbers (# of movies in a genre / # of movies)

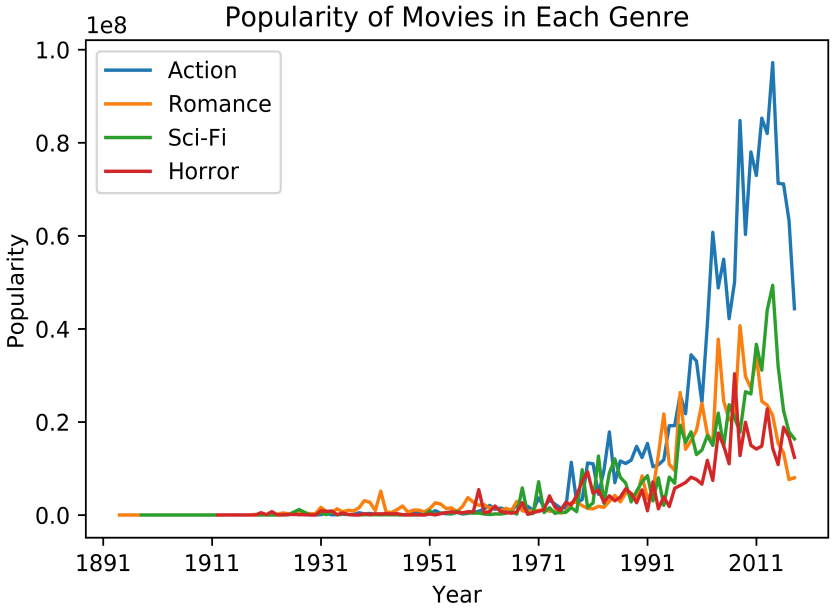


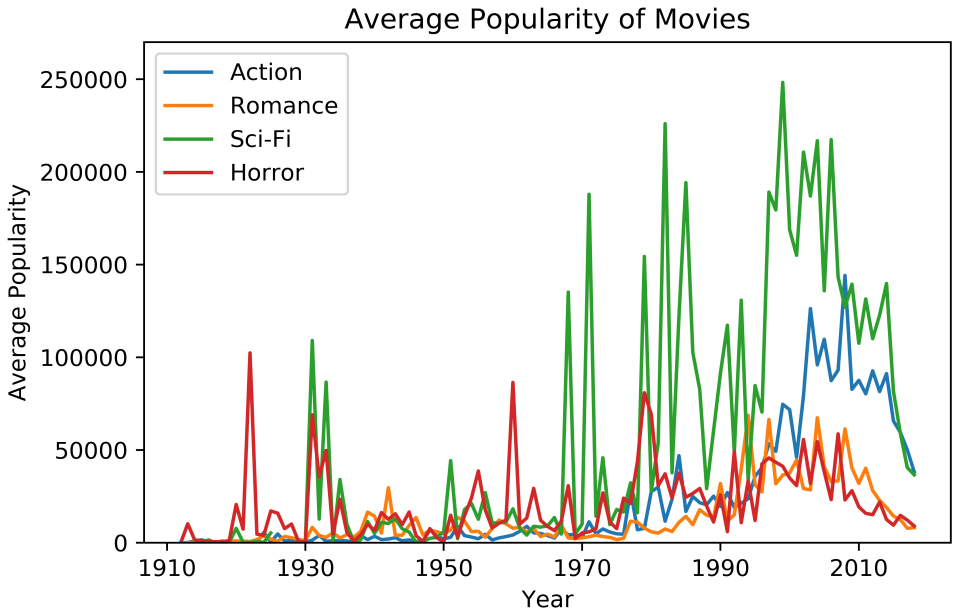




Regularization

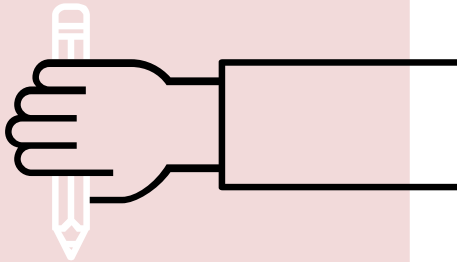
- **Problem**: More movies, higher popularity
- **Solution**: Take average (genre popularity / # of movies in that genre)



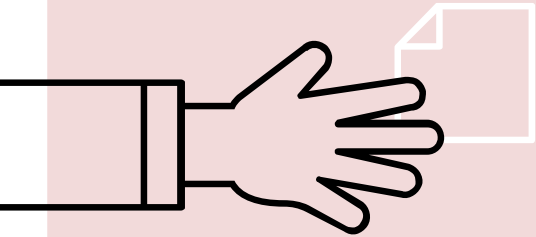


Findings

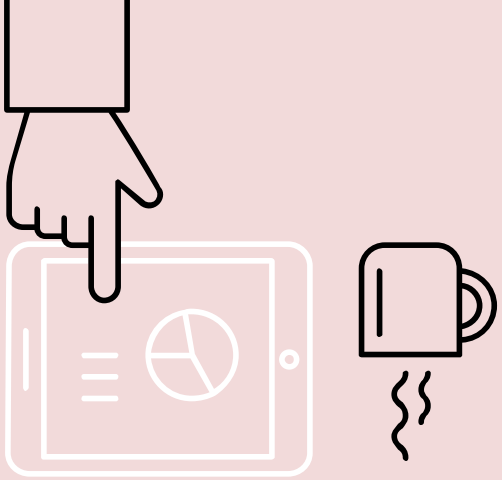
- Romance movies were more **popular back in the old days**
- People **switch their tastes** to action movies
- Producers **makes less** and less romance and action movies in recent ten years
- Producers starts to make more horror movies. However, they are not as appreciated as other genres.
- Producers make less science fiction movies, but people love the existing ones



II. Relationship Between Romance Movies and Producing Countries



Data Manipulation



Datasets

- title.akas.tsv: includes **foreign names** of every movie
- title.basics.tsv: includes **genres** of each movie
- Country_loc.tsv: contains **abbreviation, longitudes and latitudes** of each country

Key Process

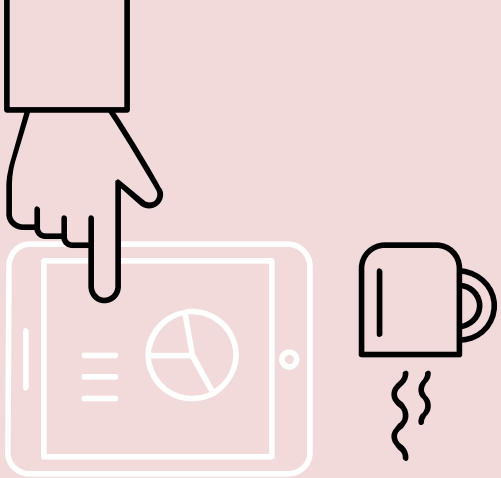
```
def filter_genre(row):
    if type(row["genres"]) is str and type(row["titleType"]) is str and "Romance" in row["genres"].split(",") and row["titleType"]=="movie":
        return True
    return False
```

- Movies: **romance** + could be classified as Movies
- Countries: Standard Alpha-2 Code + appeared in movie dataset

Data Manipulation

Datasets

- title.akas.tsv: includes **foreign names** of every movie.
- title.basics.tsv: includes **genres** of each movie
- Country_loc.tsv: contains **abbreviation, longitudes and latitudes** of each country



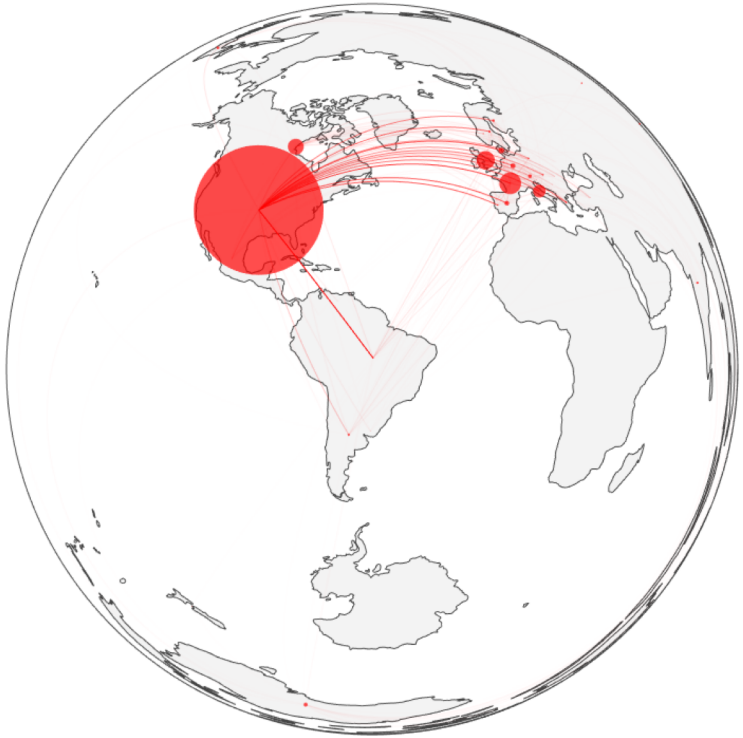
Data Manipulation

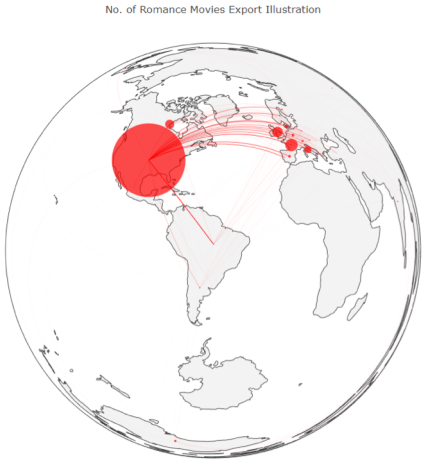
- The **number of title versions** of each movie indicates the **number of exported countries**
- Collecting all movies from the same original country to derive the relationships among countries in the world

Results

No. of Romance Movies Export Illustration

No. of Romance Movies Export Illustration





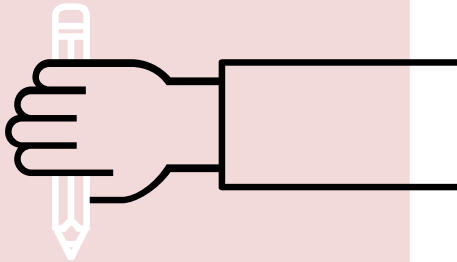
exports.html

Observations

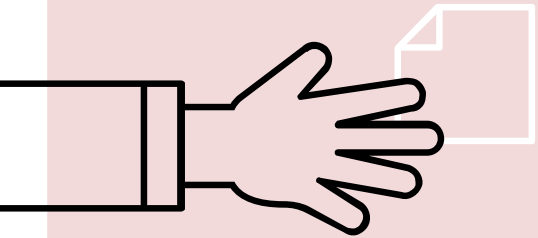
1. **Highest exporting countries** in volume:
   - U.S.
   - France
   - U.K.
   - Canada
   - Italy
2. **Dense in the Western world**, while minimal exports in the rest of the globe.

Interpretations

1. **Developed countries** produce more movies in general.
2. The concept of "Romance and Love" has **higher cultural uniformity in the Western World**, while numerous cultural implications of "love" in the East might result in few exports of their romance movies.
3. There is an enormous **gap** between Eastern and Western **understanding** of "Romance".



III. Binary Classification of Romance Movies Based on Average Rating

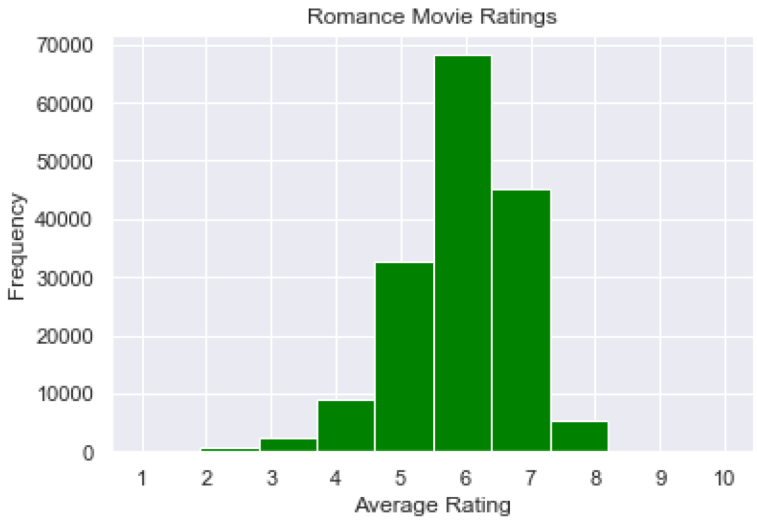


Set Up

- **Input:** $x \in \mathbb{R}^D$, D = {startYear, region, director, types}
- **Output:** $y \in \{0,1\}$

y = 1 if rating >= 6.5 "Recommended"

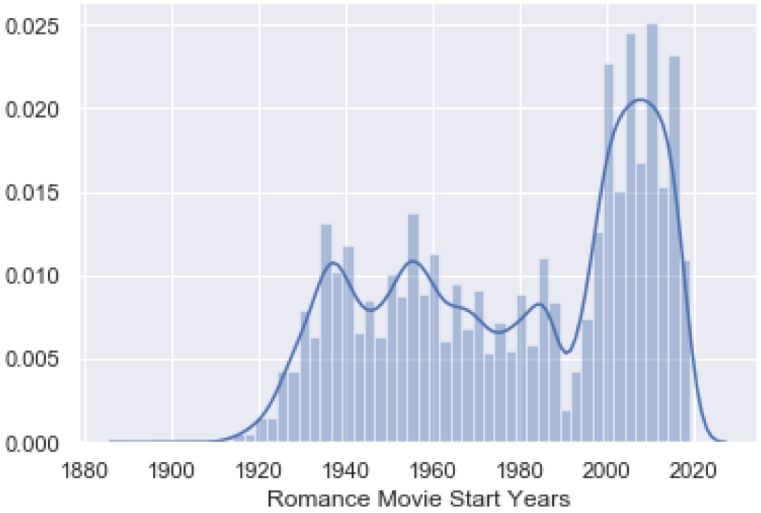
y = 0 if rating < 6.5 "Unrecommended"



Plotted using seaborn

Preprocess Data

- **Read** data from file
- Apply **clean**ing function
  - startYear: **center** the data at 1980
  - region: **eliminate** '\N'
  - types: `split(',')[0].strip()`
  - genre: only **preserve** 'Romance'
- Create a **mask**
- **Merge** features of interest into one table according to their IDs




```
# merge all data that we want to use into a single table called "movies"
movies = yearAndGenre[mask].merge(rating, on='tconst')
movies = movies.merge(regionAndType, on='tconst')
movies = movies.merge(directors, on='tconst')
```

Get Feature Matrix

- "Bag of words" model
- One-hot **encoding**
- **Concatenate** features
- Result:

  (n_samples * sum of feature dimensions) matrix

Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch

	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

Each word gets a 1x9 vector representation

```
def extract_dictionary()
Out [1]: {"original": 0}
def extract_matrix()
Out [2]: (1,0,0,...,0)
```

Get Feature Matrix -> Get Label Array

- "Bag of words" model
- One-hot **encoding**
- **Concatenate** features
- Result:

  (n_samples * sum of feature dimensions) matrix

```
def extract_dictionary()
Out [1]: {"original": 0}
def extract_matrix()
Out [2]: (1,0,0,…,0)
```

Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch

	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

Each word gets a 1x9 vector representation

Dimensionality Reduction

- Shape of feature matrix: (10000, 497)
- **Truncated Singular Value Decomposition**: need experiments!

$$A_{nxp} = U_{nxn}\, S_{nxp}\, V^{T}_{pxp}$$

$$\begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \sigma_3 & & & \\ & & & \sigma_4 & & \\ & & & & \ddots & \\ & & & & & \sigma_r \end{bmatrix}$$

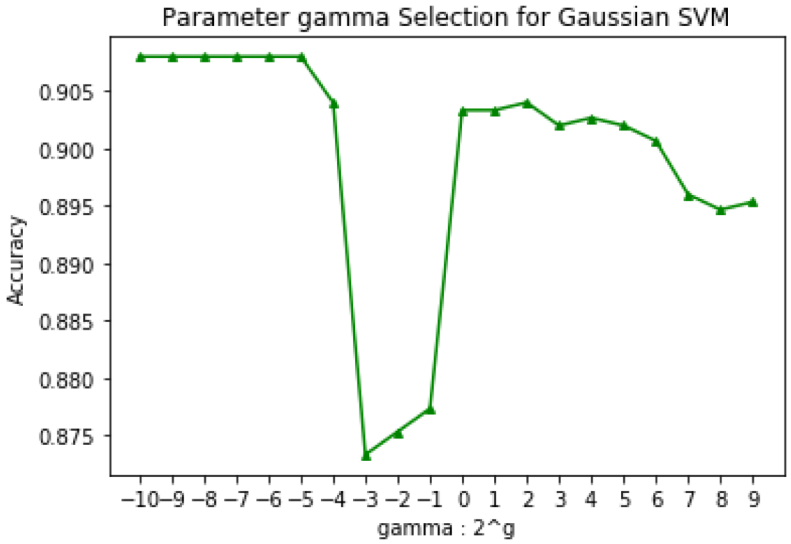
```
# apply singular value decomposition to reduce
# dimensionality for the input
tsvd = TruncatedSVD(n_components=100)
feature = tsvd.fit(feature).transform(feature)
```

Train Classifiers

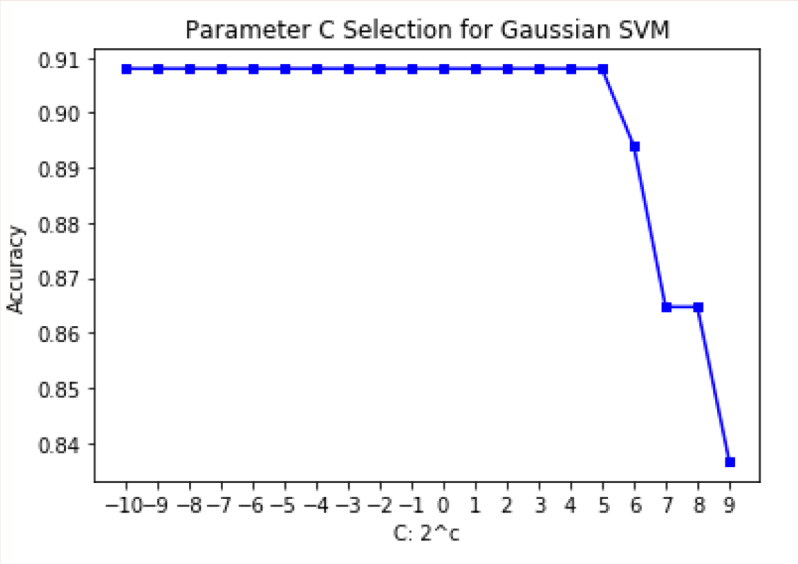
- Sample size: 10000, training size:8500

	Features	Accuracy
Dummy Classifier	uniform	0.5193
Gaussian SVM	**sklearn.StratifiedKFold sklearn.GridSearchCV**	**0.908**
Bernoulli Naïve Bayes		0.746
KNN	uniform weights/distance best k: 7 train score: 0.8267	0.8193

Best parameters for SVM: {'C': 1.0, 'gamma': 0.01}

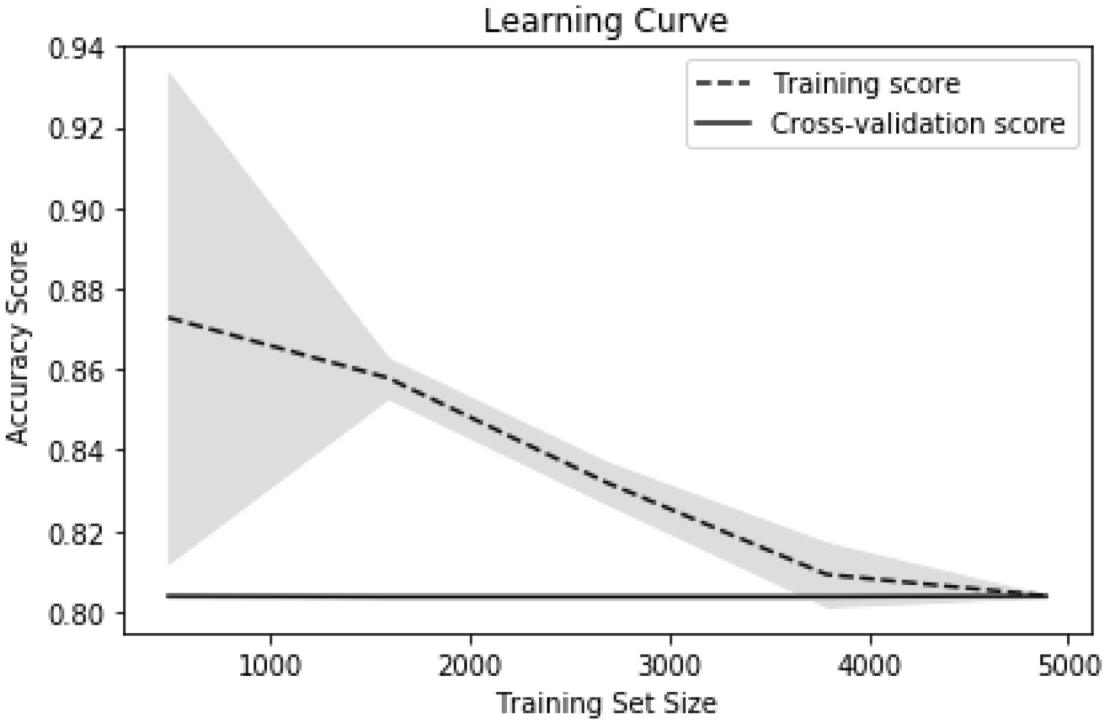


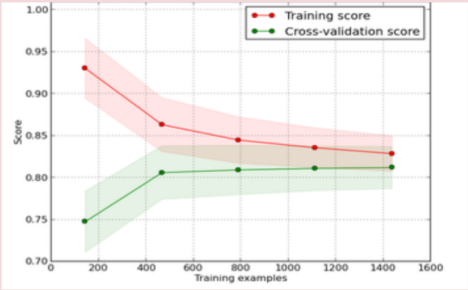
The γ parameter controls **how far the influence of a single training example can reach**. Low values mean that the influence can be far and high values mean close.



The C parameter balances between **correctly classifying** training instances and **maximizing** decision function's **margin**.

SVM Learning Curve





- Accuracy **decreases** as number of training sample increases
- Hard to **generalize**

Train Classifiers

- Sample size: 10000, training size:8500

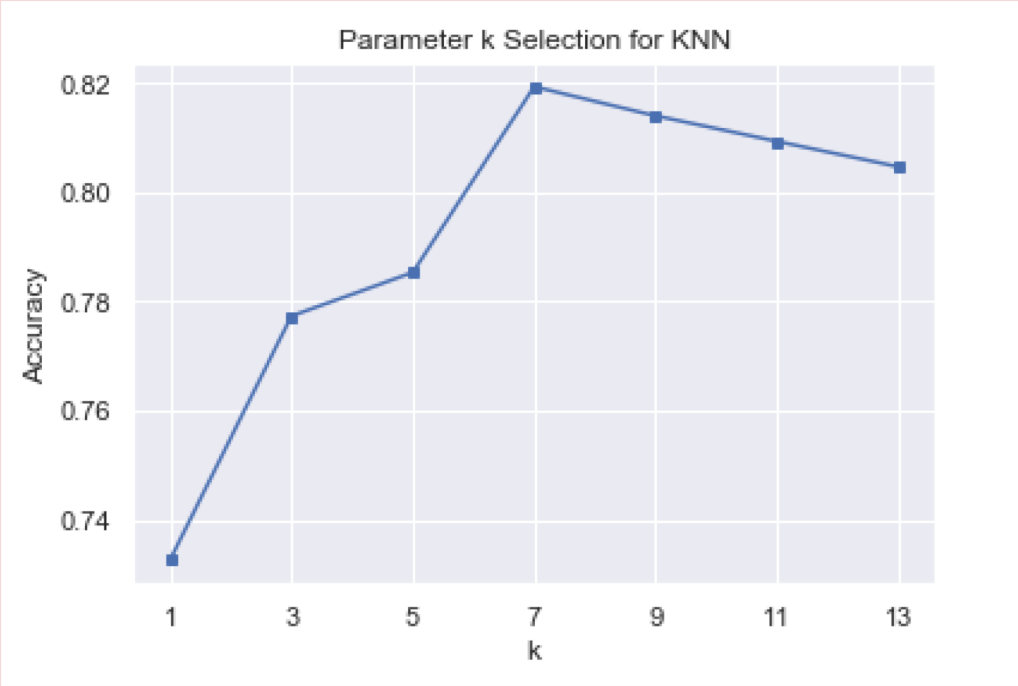
	Features	Accuracy
Dummy Classifier	uniform	0.5193
Gaussian SVM	sklearn.StratifiedKFold sklearn.GridSearchCV	0.908
Bernoulli Naïve Bayes		**0.746**
KNN	uniform weights/distance best k: 7 train score: 0.8267	0.8193

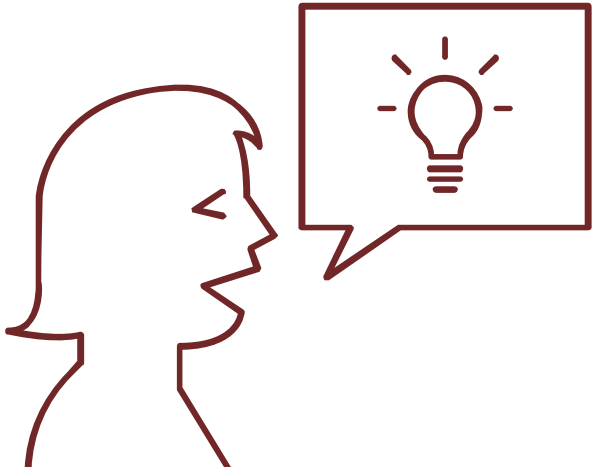
Train Classifiers

- Sample size: 10000, training size:8500

	Features	Accuracy
Dummy Classifier	uniform	0.5193
Gaussian SVM	sklearn.StratifiedKFold sklearn.GridSearchCV	0.908
Bernoulli Naïve Bayes		0.746
KNN	uniform weights/distance best k: 7 train score: 0.8267	0.8193

Best k for KNN:



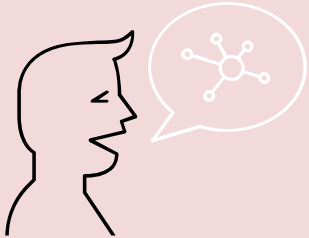


Interpretation

- **Features** are not absolutely deterministic, no strong relation
- **Models** are simple -> neural network
- Immature **data processing** -> apply further techniques in reducing noises
- **SVM** fits the best
- **Hyperparameters** are important

CONCLUSION

- As the film industry developed with time, romance movies experienced a fast development.
- Due to political, economical and cultural reasons, different countries understand romance differently and have distinct preferences.
- People have different standard/potential bias towards the quality of movies. It is hard to categorize a movie as good or bad.



Thanks for listening!